Delta College

CST 283
Project

Winter 2024
Instructor: T. Klingler

Objective

To build a complete working Java program that includes object oriented programming, processing of a large data structure, and a JavaFX user interface.

Overview & Instruction

Write a software application to manage a very large data set of health patients your organization is responsible for.

The format of the provided data file is:

```
firstName,lastName,address,city,state,zip,phone,email,date1,date2
```

Data includes general contact information. The unique key for an entry is their email. The last two fields contain the dates the patient received their COVID-19 virus innoculation (two shots are required). If one or both date files are `0000-00-00`, then they have not received it yet. Note that `date1` will of course be populated before `date2`.

Your ***back-end*** data container should include:

- A class to store and manage one `Patient` object including all fields from each data line in the file
- A binary search tree of `Patient` objects to manage all of the data (utilizing the generic BST developed in class)
- Strongly consider defining a class to "wrap around" the provided binary search tree of `Patient` objects. This will likely give more flexibility with methods called in the driver class.

You will need to include methods to add a patient, delete an entry, change patient info, and search for a given patient record. Feel free to make (minor) necessary changes to the generic binary search tree class (while maintaining its generic status).

Build a ***front-end*** set of graphical user interface that includes an interface to enable these features:

- Create an login window view that includes only login ID and password text fields (be sure the password field is hidden). If the login is for the admin (login `admin` with password `admin`), display the ***main admin*** window view

- The main interface should include options to add a patient, delete a patient, change any field in the patient record, or search for a patient. Use the patient email as the key for deletions. Include the ability to search (via the email key) to populate all of the interface fields with current data. Text fields are appropriate for most data, but integrate a drop-down list for the state selection.

- Devise an approach to include two date fields in the interface for the respective vaccines. You have flexibility in this. Two text fields will suffice but use of date choosers or another creative approach for the two calendar date storage and interface would be viewed more highly during grading. Include a basic error checking feature to be sure the first date is always less than the second date.

- Be sure to use JavaFX interface components throughout.

- Along with the basic data editing features defined above, also include options to allow the user to search for and process patients. Include the ability to:

    - For a selected state, count
        - The number having received the first shot but not the second.
        - The number not receiving either shot
        - The number receiving both shots

    - For a selected zip code, list names for
        - The number having received the first shot but not the second.
        - The number not receiving either shot
        - The number receiving both shots

- Be sure to include a ***Quit*** selection, button, or option. This should write the binary search tree back to a file and quit the application.

Finally, note that `JOptionPane` dialogs do not always "play well" with JavaFX. Please avoid `JOptionPane` dialog interface components and use JavaFX `Alert` actions instead.

Resources

Initial data file (via zipped download): patients.txt.zip

Deliverables

Demonstrate the development steps of your program with at least two version commits to the assignment Git repository.

Deliver the following to the eLearning system **Assignment Dropbox** as your final product:

- **Upload** your **source code** (.java) file(s); preferably as one zipped submission

---